

Learning Parallel Computations with ParaLab

Evgeny Kozinov and Anton Shtanyuk

Lobachevsky State University of Nizhni Novgorod
Nizhni Novgorod, Russia
{evgeniy.kozinov,ashtanyuk}@gmail.com

Abstract. In this paper, we present the ParaLab teachware system, which can be used for learning the parallel computation methods. ParaLab provides the tools for simulating the multiprocessor computational systems with various network topologies, for carrying out the computational experiments in the simulation mode, and for evaluating the efficiency of the parallel computation methods. The visual presentation of the parallel computations taking place in the computational experiments is the key feature of the system. ParaLab can be used for the laboratory training within various teaching courses in the field of parallel, distributed, and supercomputer computations.

Keywords: parallel computations · education · curriculum · numerical experiments

1 Introduction

The world of computations is becoming more and more parallel and distributed. The modern supercomputers demonstrate high computational performance of thousands teraflops ($\approx 10^{15}$ operations per second). The total number of the computational cores can reach several millions. The efficient usage of such highly developed computational facilities requires a new generation of the high-qualified professional experts.

The importance of the problem of education in the field of the parallel, distributed, and supercomputing computations (PDSC) is widely recognized by the international educational community.

One can recognize several large-scale projects, which contain the recommendations on the curricula as well as the examples of the educational courses. Among them:

- the “Curriculum Initiative on Parallel and Distributed Computing” project - carried out under NSF/IEEE-TCPP [1].
- the Computing Curricula activity implemented for several years by ACM and IEEE-CS international communities [2].
- the Russian national project on Supercomputer Education [3].

This research was supported by the Russian Science Foundation, project 15-11-30022 “Global optimization, supercomputing computations, and applications”.

Some important results in this direction have been achieved also within the framework of SIAM-EESI project on the development of education in the field of computational science and engineering [4]. Some other results are presented in [5-8].

In the paper, we are going to focus on the issues of providing a necessary laboratory training along with the wide spectrum of the problems, arising in the education in the PDSC field. Thus, in order to conduct the laboratory training, it is necessary to provide access to a real supercomputer system (preferably, even to several various supercomputers with different hardware and architectures). Computational experiments may take quite a long time and, therefore, may require large computational resources (including the financial ones). Finally, the conducted parallel computations are not observable visually: the developers of the parallel algorithms and programs cannot see, which processors execute distributed computations, what data are transferred, and which processors are involved in such transfer, etc.

All the issues listed above essentially complicate learning in the PDSC field. These difficulties can be reduced by means of development and wide use of the educational software systems. Such systems can visualize the key aspects of the parallel, distributed, and supercomputer computations - the most difficult ones to understand.

In this paper, we present the Parallel Laboratory (ParaLab) teachware system [16], which provides the capabilities to carry out the computational experiments for the purpose of learning and investigation of the parallel algorithms for solving complex computational problems. The system can be applied in the laboratory training within various educational courses in the PDSC field, giving the learners an opportunity:

- to simulate the multiprocessor computational systems with various processor number and network topologies,
- to visualize the computation processes and the data transfer operations taking place in the parallel solving of various computational problems,
- to evaluate the efficiency of the studied parallel computation methods.

The paper goes as follows: Section 2 contains a general description of the system. Section 3 describes the parallel computational methods that can be studied with ParaLab. In section 4, a set of laboratory works based on ParaLab is given. Section 5 concludes the paper.

2 Related Work

Many researchers and teachers pay a considerable attention to the development of the tools for visualization of the algorithms and programs and to wide use of them in education. One of the first reviews of the methods for visualizing the parallel programs was presented in [9]. The estimate of the efficiency of animation and visualization tools was considered in [10-12]. Some existing visualization tools were considered in [13-14]. The capabilities of the Matlab system are used widely for the visualization of the algorithms and programs [15].

3 ParaLab Overview

In general, ParaLab is an integrated software environment for learning and research of the parallel algorithms for solving complex computational problems.

A wide range of the tools for visualizing the parallel computations and for analyzing the experimental results allows studying the efficiency of various algorithms for different computational systems, making the conclusions on the scalability of the parallel algorithms, and evaluating the possible speedup of the parallel computations.

The main feature of the system is that the parallel computations are performed in the simulation mode, and therefore, studying the parallel methods can be performed on any ordinary computer. To evaluate the required characteristics of parallel computations (execution time, speedup, efficiency, etc.) the appropriate theoretical models are used [17-19].

ParaLab provides the following capabilities for studying parallel computations.

1. **Simulating the computational system.** To simulate a computational system, one can define the topology of a parallel computational system for carrying out the computational experiments, select the number of processors in this topology, set the performance of the processors, select the communication network parameters and the communication method (see Figure 1). Within the framework of the system, the support of several standard topologies is provided, including the line (farm), the ring, the star, the mesh, the hypercube, and the complete graph (clique) ones.

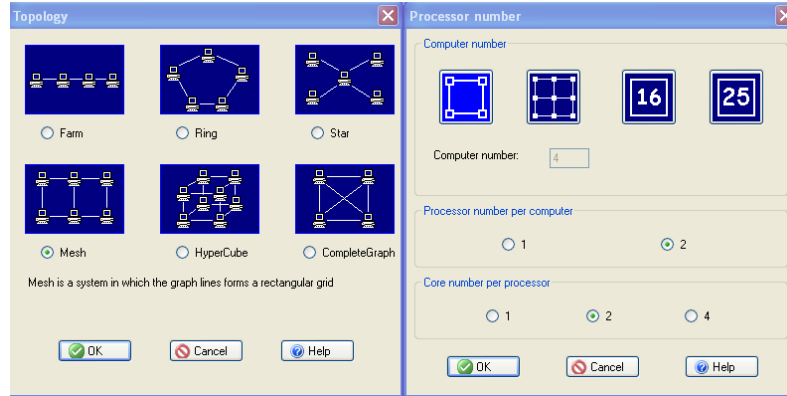


Fig. 1. Dialog windows for setting the system parameters

ParaLab allows to simulate high-performance computational systems that can consist of a set of computational nodes. Each computational node can contain one or several processors, and each processors can have one or several cores.

2. **Selecting the problem statement and the method for its solving.** Within the framework of the ParaLab system, the student can perform the computational experiments for the following set of problems: matrix-vector multiplication, matrix multiplication, solving the systems of linear equations, sorting, graph processing, solving the differential equations in partial derivatives, and multidimensional global optimization.
3. **Performing a computational experiment.** Prior to execution of a computational experiment, one can set up the necessary visualization parameters, select the desired demonstration rate, the visualization mode of the data transfer between the processors, and the granularity degree of the visualization of the parallel computations performed. ParaLab provides a wide choice of tools for carrying out the computational experiments. The experiments can be performed either in the automatic mode or in the step-by-step mode, when calculations are suspended after each iteration of the algorithm is completed. It should be noted that several different experiments with various types of the multiprocessor systems, problems, or parallel methods can be run simultaneously in time-sharing mode.

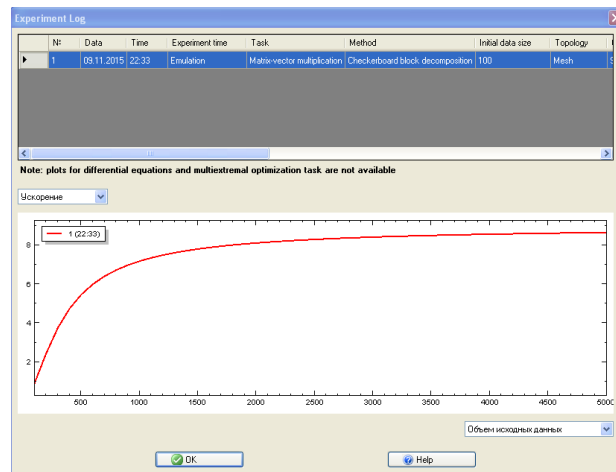


Fig. 2. The experiment log window

4. **Analyzing the results of the computational experiments.** The ParaLab system accumulates the results of the computational experiments automatically. The system provides the tools for plotting the dependencies featuring the parallel computations (execution time, speedup, efficiency) vs the parameters of the problem or the computational system. The dependencies are plotted according to the theoretical models for the computational complexity of the parallel algorithms (Figure 2). An example of the visual presentation of the parallel computations in solving the problem of the matrix multiplication by the parallel algorithm with the

block-striped matrix decomposition is presented in Figure 3. Two windows for performing the computational experiments were open in the workspace of the ParaLab system. The computational systems consisting of 9 nodes were used for both computational experiments where each node contains 2 two-core processors. The mesh topology was used in the system shown in the left window and the complete graph (clique) one in the system shown in the right window.

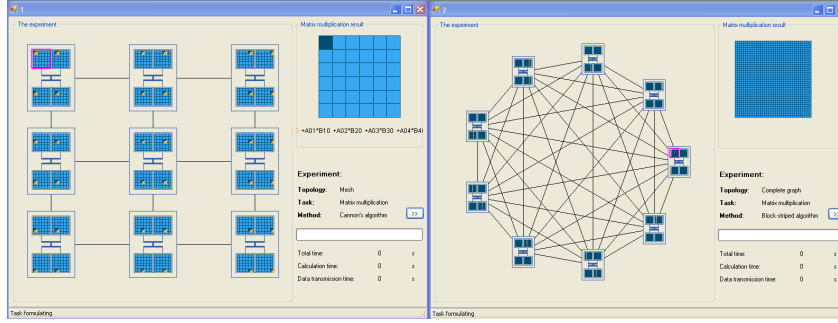


Fig. 3. Visual representation of the numerical experiments

The parameters of the problems being solved, such as the problem name, the selected parallel method, and the initial data volume are shown in the “Experiment” list at the lower right side of the window. In the “Topology” list, the attributes of the selected computational system, namely the topology, the number and performance of the processors, and the network parameters are listed.

Below the “Experiment” field, the bar indicator shows the progress of the algorithm execution. The “Total time” and “Communication time” boxes show the execution time of the parallel algorithm.

4 Parallel Methods for Studying with ParaLab

A wide choice of the parallel methods for solving a number of the problems of computational mathematics can be studied with ParaLab [16-19].

1. **Matrix computations.** The matrix computations are used for solving numerous scientific and technological problems. Incurring rather high computational costs, the matrix computation methods are a good example for learning various methods of parallel computations.

The following algorithms are provided by ParaLab:

- parallel algorithms of the matrix-vector multiplication with the block-striped and checkerboard block matrix decomposition,

- a parallel algorithm of the matrix multiplication for the block-striped data decomposition scheme and two parallel methods (the Fox and Cannon algorithms) for the checkerboard block matrix decomposition,
 - the parallel Gauss method for solving the systems of linear equations.
2. **Data sorting.** Sorting is one of classical data processing problems. It is a problem of arrangement of the elements of a non-ordered dataset in the monotonous ascending or descending order. A parallel variant of the bubble sorting method and the parallel Shell and quick sorting algorithms are implemented in ParaLab.
 3. **Graph processing.** Data representation in the form of graphs is used widely in modeling of various phenomena, processes, and systems. Therefore, the graph processing is applied in the practical applications widely. For the graph processing problems, ParaLab uses the Prim's parallel method for finding the minimum spanning tree and the Dijkstra's and Floyd's algorithms for finding the shortest paths.
 4. **Solving the differential equations in partial derivatives.** The differential equations in partial derivatives are a widely used approach applied for mathematical modeling in various fields of science and technology. The amount of computations required for the numerical solving of the differential equations is usually large, and utilizing the high performance computational systems is traditional for this field of computational mathematics. In the ParaLab system, the parallel Gauss-Seidel method is implemented for this class of problems.
 5. **Multiextremal optimization.** The optimization problems describe how to select the best variants, while developing novel devices, objects, and systems and, therefore, have found an ultimately wide application in various fields of the human activities. The multiextremal (global) optimization problems, which assume several local optima in the search domain, belong to the most complex optimization ones. The parallel index method [20-21] is implemented for solving the multiextremal optimization problems in ParaLab.

5 Laboratory Training with ParaLab for Learning Parallel Methods

ParaLab is designed for research and studying the parallel algorithms for solving complex computational problems. ParaLab can be used by the university students and teachers within the framework of the laboratory training in various educational courses in the PDSC field. ParaLab system can be applied also in research for evaluating the efficiency of the parallel computations as well.

The laboratory training with ParaLab system can be implemented in accordance with the following successive scheme:

- Simulating the multiprocessor computational systems (the topology selection, setting the number and performance of processors, selection of the data transfer method, and setting up the communication network parameters);

- Choosing the class of the problems to be solved and setting the problem parameters;
- Selecting the parallel method for solving the problem and setting the values of its parameters;
- Setting up the graphical indicators for visualizing the parallel computation process (the status of data on the system processors, the data transfer via the network, the current computational results);
- The execution of the experiment in the computations simulation mode; choosing the experiment mode: automatic or step-by-step, single or a series of executions, single or multiple experiments in the time-sharing mode for different variants of the computational system topologies, number of processors, the problem parameters, etc.;
- Analyzing the experimental results accumulating in the experiment log file; evaluating the execution time subject against the complexity of the problem and the number of processors; plotting the dependencies of the speedup and the efficiency of the parallel computations;
- Implementing the experiments with the real parallel computations; execution of the parallel programs on a single processor and on a multiprocessor computational system using the remote access; comparing the theoretical estimates with the results of real computational experiments. The laboratory training with ParaLab can be conducted, for example, according to the following set of assignments:
- A student solves a complex computational problem using several parallel methods and a computational system, compares the results to each other, and interprets them within the theory of the parallel algorithms;
- A student constructs several computational systems in such a way that allows to demonstrate the basic theoretical concepts of parallel computations;
- A student constructs one or several computational systems and solves the problems with various values of the computational system parameters, thus studying the effect of the parameters on the time of the algorithm execution;
- A student performs real computation experiments using a cluster in the remote access mode and compares the results of real and simulated experiments.

In practical application of the ParaLab system for teaching parallel computations, the following scheme of the laboratory training could be recommended.

Lab 1. Simulating a computational system. This lab is aimed at studying the architecture of the multiprocessor systems. Using ParaLab, standard topologies of the computational systems can be considered with the possibility of the visualization of them at various number of the computational nodes, processors, and cores. Within the framework of the lab, the communication network performance (latency and bandwidth) as well as the basic methods of data transfer (message and packet modes) can also be studied.

This lab could be recommended within the framework of the laboratory training for studying the “Architecture” section of the recommended curriculum developed within NSF/IEEE-TCPP the “Curriculum Initiative on Parallel and

Distributed Computing” project (the “Architecture of Computational Systems” training course).

Lab 2. Studying the parallel methods of the matrix computations. Within the framework of this lab, the basic methods of matrix distribution between the processors (the horizontal and vertical block-striped schemes, the checkerboard block decomposition of the matrices) can be considered.

Also, the problem of numerical solving the systems of linear equations can be considered as an additional topic for this lab.

This lab can be recommended within the framework of the laboratory training for studying the “Algorithms” section of the recommended curriculum, developed in the framework of the NSF/IEEE-TCPP “Curriculum Initiative on Parallel and Distributed Computing” project (the “Parallel Programming” and “Numerical Methods of Parallel Computations” training courses).

Lab 3. Studying the parallel data sorting methods. This lab continues the topic of studying the parallel methods for solving the complex computational problems. Within the framework of this lab, the parallel bubble sorting algorithm, the Shell sorting method, and the quick sorting algorithm can be considered.

Extended utilizing of the tools available in ParaLab for visual presentation of the parallel computation process is assumed within the lab. Prior to executing the computational experiments, the demonstration rate and the modes of demonstration of the data transfer operations can be changed. Also, the step-by-step mode of execution of the algorithm iterations can be activated. It is useful to visualize the calculations performed by one of the processors in a separate window as well.

This lab can be recommended within the framework of the laboratory training for studying the “Algorithms” section of the recommended curriculum developed within the framework of the NSF/IEEE-TCPP “Curriculum Initiative on Parallel and Distributed Computing” project (the “Parallel Programming” and “Numerical Methods of Parallel Computations” training courses).

Lab 4. Studying the parallel methods of graph processing. Within the framework of this lab, the Prims parallel algorithm for finding the minimum spanning tree and the Dijkstra’s parallel method for finding the shortest paths are studied. The graphs for performing the experiments are generated by a random graph generator or can be set by a graphic editor by uploading from a file.

Conducting the lab can be performed in the manner similar to the labs 3 and 4.

The labs for studying the parallel methods can be extended by the topics considering the parallel numerical methods for solving the differential equations in partial derivatives and the global optimization problems.

Lab 5. Studying the methods for analyzing the experimental results. The lab is assigned for studying the basic principles of carrying out the computational experiments and the methods of accumulation and analysis of the obtained experimental data. For studying this topic, ParaLab accumulates the results of the performed computations in the experimental data log file. Within the framework

of this lab, the computational experiments are performed, the numerical results stored in the experimental log file should be analyzed, and the dependencies of the execution time (or speedup) on the problem (the amount of the initial data) and of the computational system parameters (the number of processors, nodes, cores) can be plotted.

This lab can be recommended within the framework of the laboratory training for the “Computational Experiments and Methods of Experimental Data Analyzing” training course.

6 Conclusions

In this paper, we present the ParaLab teachware system, which can be applied for studying the methods of parallel computations. ParaLab provides the tools for modeling the multiprocessor computational systems with various topologies of the data transfer network, for performing the computational experiments in the simulation mode, and for evaluating the efficiency of the parallel computation methods being studied. The visual demonstration of the parallel computation processes executed during the performed computational experiments is the key feature of ParaLab system.

The system can be applied for the laboratory training within various educational courses in the PDPS field. ParaLab is applied intensively in the educational activities at University of Nizhny Novgorod as well as in other Russian universities.

ParaLab is presented on the website of the Supercomputing Technologies Center, Lobachevsky State University of Nizhni Novgorod (see <http://www.hpcc.unn.ru/?doc=107>) as a part of the set of educational resources.

7 Acknowledgments

This research was supported by the Russian Science Foundation, project 15-11-30022 “Global optimization, supercomputing computations, and applications”.

References

1. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing at <http://www.cs.gsu.edu/~tcpp/curriculum/>
2. Computing Curricula Computer Science 2013 at <http://ai.stanford.edu/users/sahami/CS2013/>
3. Voevodin V., Gergel V. Supercomputing Education: The Third Pillar of HPC at http://www.exascale.org/mediawiki/images/3/3c/ECSS_2010.pdf
4. Future Directions in CSE Education and Research Report from a Workshop Sponsored by the Society for Industrial and Applied Mathematics (SIAM) and the European Exascale Software Initiative (EESI-2) at <http://wiki.siam.org/siag-cse/images/siag-cse/f/ff/CSE-report-draft-Mar2015.pdf>

5. Computer science in Parallel (CSinParallel) - <http://serc.carleton.edu/csinparallel/index.html>
6. A Survey on Training and Education Needs for Petascale Computing - www.prace-ri.eu/IMG/pdf/D3-3-1_document_final.pdf
7. Rague, B. Teaching parallel thinking to the next generation of programmers // Journal of Education, Informatics and Cybernetics, vol. 1, no. 1, pp. 4348, 2009
8. Gergel, V., Liniov, A., Meyerov, I., Sysoyev, A. NSF/IEEE-TCPP Curriculum Implementation at the State University of Nizhni Novgorod // IPDPSW '14 Proceedings of the 2014 IEEE International Parallel & Distributed Processing Symposium Workshops. IEEE Computer Society Washington, DC, USA, 2014, pp. 1079-1084.
9. Kraemer, E., Stasko, J.T. The visualization of parallel systems: an overview // Journal of Parallel and Distributed Computing, 18,1993, 105117.
10. Hundhausen, C.D., Dougla, S.A., Stasko J.T. A Meta-Study of Algorithm Visualization Effectiveness // Journal of Visual Languages & Computing, 13(3), 2002, 259290.
11. Urquiza-Fuentes, J., Velzquez-Iturbide, J. Towards the effective use of educational program animations: the roles of students engagement and topic complexity. Computers & Education, 67, 2013, 178192.
12. Lazaridis, V., Samaras, N., Sifaleras, A. An empirical study on factors influencing the effectiveness of algorithm visualization // Comput. Appl. Eng. Educ. 21, 2013, 410420.
13. Ben-Ari, M., Bednarik, R., Ben-Bassat, L.R., Ebel, G., Moreno, A., Myller, N., & Sutinen, E. A decade of research and development on program animation: the Jeliot experience // Journal of Visual Languages & Computing, 22(5), 2011, 375384.
14. Sorva, J., Karavirta, V., & Malmi, L. A review of generic program visualization systems for introductory programming education // ACM Transactions on Computing Education (TOCE), 13(4), 2013, 15.
15. Teaching with Data, Simulations and Models. Topical Resources at http://serc.carleton.edu/NAGTWorkshops/data_models/toolsheets/MATLAB.html
16. Gergel, V., Labutina, A. The ParaLab system for investigating the parallel algorithms // Lecture Notes in Computer Science, 2010, 6083 LNCS, pp. 95-104
17. Kumar V., Grama A., Gupta A., Karypis G. Introduction to Parallel Computing. - The Benjamin/Cummings Publishing Company, Inc., 1994.
18. Quinn, M.J. Parallel Programming C with MPI and OpenMP. - Mccraw-Hill, New York, 2004.
19. Gergel, V.P. Theory and Practice of Parallel Computations. BINOM, Moscow, 2007. (In Russian)
20. Strongin, R.G., Sergeyev, Ya.D. Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms. Dordrecht : Kluwer Academic Publishers, 2000.
21. Gergel, V.P., Strongin, R.G. Parallel computing for globally optimal decision making // Lecture Notes in Computer Science, 2763, 2003, pp. 76-88.